

## INTRODUCTION TO SIMULATION USING INTELLIGENT SIMULATION INTERFACE (ISI)

Joseph M. Fegan

Gearóid M. Lane

Paul J. Nolan

ExTech Limited  
 3 Buttermilk Walk  
 Galway, IRELAND

Department of Mechanical Engineering  
 University College Galway  
 Galway, IRELAND

### ABSTRACT

ISI is a graphical user interface for general purpose simulation languages. Currently it provides an interface to the SIMAN language, although other languages such as SLAM and GPSS could also be used. ISI allows a hierarchical modeling approach which satisfies the often conflicting requirements of ease-of-use and functionality. By providing graphical model building, hierarchical modeling, interactive experiment definition, instant animation and graphical output processing, ISI transforms a general purpose language into a complete simulation environment.

### 1 INTRODUCTION

In recent years a great number of software tools for discrete event simulation have become available. The majority of these tools can be classed as either *general purpose simulation languages* or *special purpose simulators* (Law and McComas 1990). General purpose simulation languages can be used in a wide variety of problem domains and generally offer a high level of flexibility in modelling. This flexibility means that they can cope easily with systems which contain non-standard components. However, their use is often restricted due to the considerable amount of expertise, time and effort needed to build a model (Shannon et al. 1985). Some of the most popular general purpose simulation languages are SIMAN (Pegden et al. 1990), SLAM (Pritsker 1986) and GPSS (Schriber 1990).

In contrast, simulators are generally aimed at allowing models to be developed and tested quickly. However, their usefulness is severely restricted by their lack of flexibility and their ability to model a given situation at only a single level of detail. WITNESS (Murgiano 1990), SIMFACTORY (Goble 1990) and AUTOMOD II (Norman 1990) are

examples of special purpose simulation systems and are designed to be used to model manufacturing processes.

A simulation system which combines the modelling power of a general purpose simulation language with the ease of use and rapid model development of a special purpose simulator would provide an ideal environment for modelling. Some of the desirable features of such a system are given in Table 1. This table also shows which of these features are generally available in special purpose simulation systems and general purpose simulation languages.

Table 1: Simulation System Features

| Desired Feature              | Ideal System | Special Purpose | General Purpose |
|------------------------------|--------------|-----------------|-----------------|
| Ease of Use                  | ✓            | ✓               | ✗               |
| Fast Model Development       | ✓            | ✓               | ✗               |
| Fast Verification/Validation | ✓            | ✓               | ✗               |
| Integrated Environment       | ✓            | ✓               | ✗               |
| Automated Animation          | ✓            | ✓               | ✗               |
| Ease of Communication        | ✓            | ✓               | ✗               |
| Broad Application Domain     | ✓            | ✗               | ✓               |
| Non-Standard Components      | ✓            | ✗               | ✓               |
| Various Abstraction Levels   | ✓            | ✗               | ✓               |
| Reusable Components          | ✓            | ✗               | ✗               |
| Team Work                    | ✓            | ✗               | ✗               |
| Transfer of Expertise        | ✓            | ✗               | ✗               |

The concept for ISI evolved from a joint university/industry R&D project about the requirements for use of simulation in industry. Based on feedback from existing and potential users of simulation (Nolan et al. 1988), it became clear that although a special purpose simulator is useful as a "first-cut" in many applications, a simulation tool which combines the flexibility of a general purpose

simulation language with the manufacturing system features and user interface of a special purpose simulation language would be ideal for detailed simulation studies.

## 2 ISI FEATURES

ISI is a total simulation support environment for SIMAN. It provides a graphical user interface to SIMAN which supports model building, editing of model parameters, experiment definition, code generation, run-time animation and post-processing. ISI is designed to be both user-friendly and highly efficient. The structure of ISI is summarized in Figure 1. ISI is described in greater detail by Lane et al. (1989).

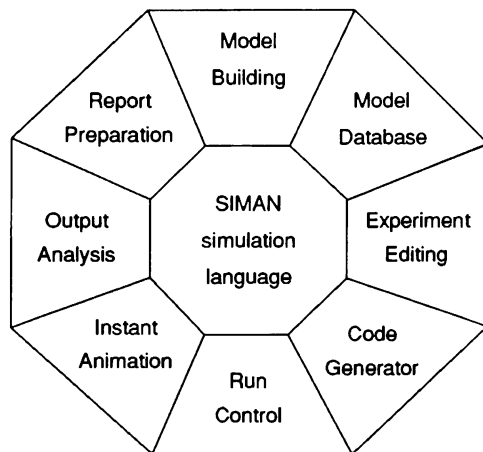


Figure 1: ISI Overview

Some of the most significant features of ISI are listed below:

**Integration.** All steps of the simulation project can be carried out without leaving ISI. Facilities for model building, data editing, experiment definition, code generation, animated execution, results processing and report preparation are all supplied as modules of ISI.

**Ease-of-use.** ISI has been designed with both non-experts and professionals in mind. Interactive graphics and pull-down menus simplify the model development task. A CAD-style interface with unlimited zooming and panning makes models of any size easily accessible. A hypertext-based help facility with over 500 screens of help is provided.

**Flexibility.** ISI is fully compatible with SIMAN, including the extensive new features added in SIMAN IV. Because it uses the powerful SIMAN language, ISI is not restricted to any one class of problems. ISI has been applied with great success to a wide range of application areas, such as:

manufacturing (Fegan and Nolan, 1991), banking and service industries (Miles and Lane, 1991), emergency planning (Lane, Nolan and Fegan 1991), mining and chemical engineering.

**Hierarchical Modelling.** Libraries of special purpose building blocks can be synthesized hierarchically from the SIMAN building blocks. Using these high level building blocks, non-expert or infrequent users of simulation can rapidly and accurately build simulation models. Thus ISI can be customized to different application areas. This very powerful feature is discussed in more detail in the next section.

**Graphical Modelling.** Model building uses a powerful menu-driven network editor. All of the other stages of modeling use this graphical network. In particular, animated model execution needs no further set-up or artistry.

**Animation.** ISI uses the same graphical model developed in model building to provide instant animation. Moving icons show the progress of entities through the model. Barcharts monitoring the accumulation of entities in queues are automatically displayed. Each block in the network is continuously updated to show its status.

**Output Processing.** ISI's menu driven output processor allows SIMAN output files to be plotted in a variety of business graphics formats, or exported to a number of spreadsheets and graphics packages. Superimposed plots and multiple windowing allow different scenarios to be compared graphically.

## 3 HIERARCHICAL MODELING

ISI allows models to be developed hierarchically. Hierarchical modelling denotes the ability to develop new functionality based on existing constructs within the simulation system. Several existing "low-level" constructs are grouped together to form a new higher level construct. This new approach is based largely on the principles of object oriented programming (Cox 1986). The object oriented approach is built on the concept of classes and instances. A *class* is a grouping of objects which have similar properties. Each member of a class is termed an *instance* of that class. Classes themselves may be instances (subclasses) of more general classes (superclasses), thus providing a hierarchical structure. The use of classes in object oriented programming allows a high level of modularity in that each class may be built and tested independently.

This hierarchical approach can also be used to advantage in viewing physical systems. In the case of manufacturing systems, machines, conveyors and transporters which are found at various points in a facility can be viewed as instances of particular classes. A class would exist for each type of machine. Different machine types with similar properties (e.g. different types of lathes) could be grouped into superclasses. Similarly, production cells, flow lines etc. could be viewed as superclasses containing networks of instances of subclasses. ISI's ability to take advantage of this inherent hierarchy offers tremendous advantages over conventional systems. Among the advantages of the hierarchical modelling approach are:

**Repetition:** A given subsystem may occur a number of times in the same model. This subsystem need only be modelled once in a hierarchical system.

**Reusable Components:** Libraries of subsystem models developed for one simulation project may be used again in further projects. Each subsequent project becomes easier, as the modelling team may derive maximum benefit from previous experience.

**Modularity:** A modelling project may be divided into a number of sub-tasks so that model components can be built and tested independently. This is particularly advantageous when a modelling team is working on the project.

**Model Changes:** The graphical modelling approach and modular nature of the model facilitate updating the model to reflect changes in the real world situation. This increases the useful lifetime of the model.

**Debugging/Validation:** Since each component can be developed and tested independently, model debugging and validation is greatly expedited.

**Communication:** The operation of a hierarchical model can be viewed at various levels of detail. This allows the operation of the model and the results to be presented at any level.

#### 4 EXAMPLE

Consider the schematic diagram of part of a manufacturing process given in Figure 2. Examining this system from a hierarchical viewpoint, it can be seen that it contains a repeated sub-process consisting of a machine followed by a quality

control operation at which the processed workpiece is either passed, failed or queued for reworking. The subprocesses differ from each other in that they use different machines and operators and have different cycle times for the machines and different pass/rework/fail ratios.

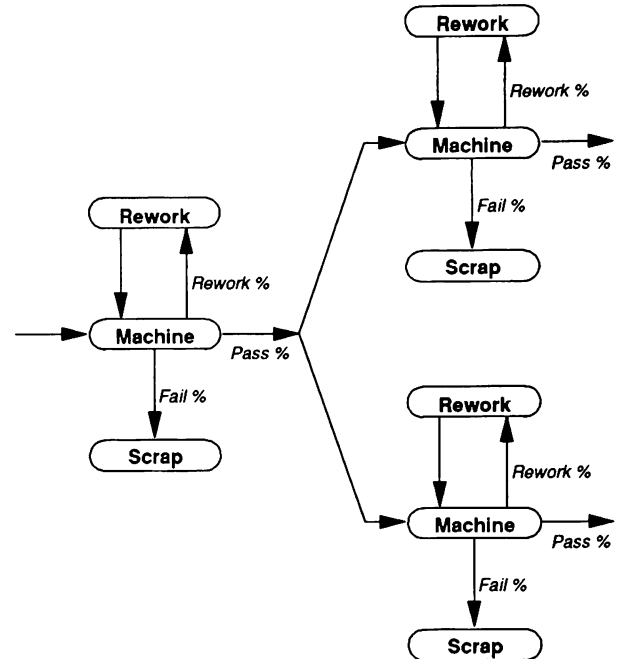


Figure 2: Manufacturing Process

Using the ISI macro facility, a macro model representing this sub-process can be developed. The procedure is outlined in Figure 3. First, the graphical sub-model shown in the figure is developed using the graphical model building facilities of ISI. This model is then "shrunk" into a macro using the ISI macro definer and a graphical icon is designed to represent this new building block.

When defining the macro, those parameters of the sub-model which are to be parameters of the macro are also defined. In this case, the parameters of the macro would be the name of the machine to be used, the name of the operator, the cycle time and the probabilities of a workpiece being passed, sent for reworking or scrapped. When the macro is complete it is stored in a macro library for later use.

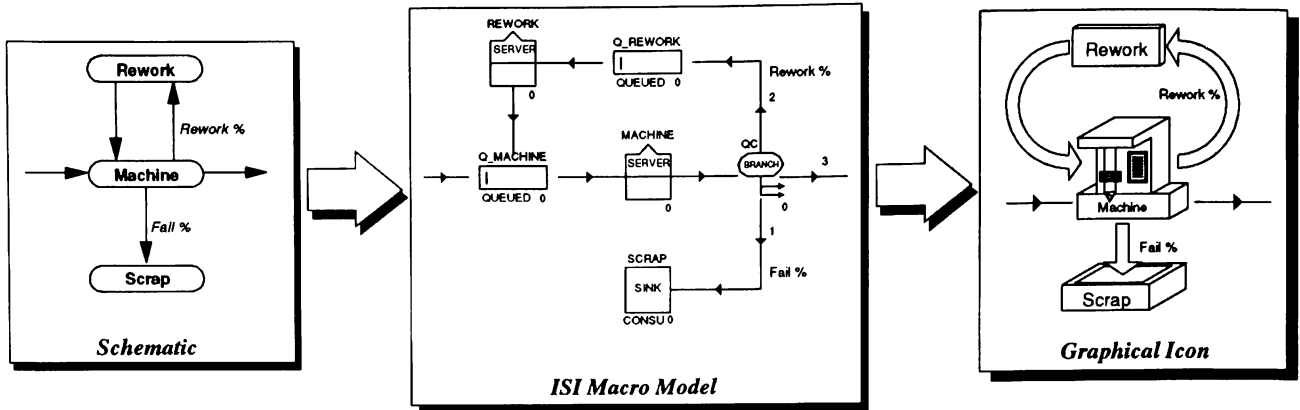


Figure 3: Macro Modelling

Once this macro for the repeated sub-process has been defined, a model for the complete system shown in Figure 2 can be constructed very easily. This model contains three instances of the new macro, one for each sub-process. The overall system model constructed using this macro is shown in Figure 4.

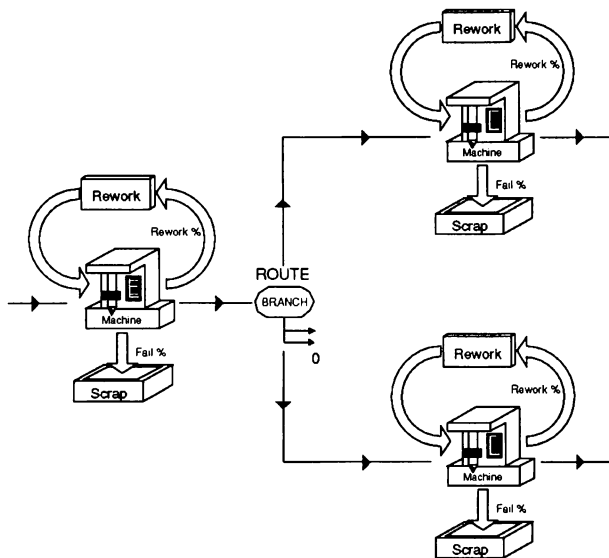


Figure 4: Final System Model

In a real-world application, the macro library would typically contain many more macros. A manufacturing library could, for example, contain macros for components like machines (lathes, milling machines) material handling devices (conveyors, transporters, AGVs) and other manufacturing equipment and processes which occur commonly. In fact, a macro library generally contains a set of macros which can be used to develop models in a particular application area quickly and easily.

### 5 CONCLUSIONS

In this tutorial, the ISI interface to SIMAN has been presented. The hierarchical features of the ISI macro facility have been outlined. The example presented has shown how an overall system model can be developed by defining new building blocks to represent repeated subprocesses.

### REFERENCES

Cox, B., *Object Oriented Programming*, Addison-Wesley, Reading, MA, 1986.

Fegan J.M. and P.J. Nolan, Building Special Purpose Manufacturing System Simulators using the ISI Macro Facility, in *Proceedings of ICCIM '91, the International Conference on Computer Integrated Manufacturing*, GINTIC Institute of CIM, Singapore, Oct. 2-4, 1991.

Goble, J., Introduction to SIMFACTORY II.5, in *Proceedings of 1990 Winter Simulation Conference*, eds. O. Balci, R.P. Sadowski and R.E. Nance, New Orleans, LA, SCS, San Diego, CA, pp136-139, Dec. 9-12, 1990.

Lane, G.M., Fegan, J.M, Nolan, P.J. and Flynn, J., A Framework for the Hierarchical Representation of Discrete Event Simulation Models using Macros, in *Proceedings of the Third European Simulation Congress*, Edinburgh, Scotland, SCS Europe, Ghent, Belgium, 262-267, Sept. 5-8, 1989.

Lane, G.M., P.J. Nolan and J.M. Fegan, Simulation of the Emergency Evacuation of Complex Buildings using ISI, in *Proceedings of Building Simulation '91, the 2nd World Congress on*

- Technology improving the Energy Use, Comfort, and Economics of Buildings Worldwide*, IBPSA, Nice, France, August 1991.
- Law, A.M., and M.G. McComas, 1990. Secrets of Successful Simulation Studies. *Industrial Engineering* 22, no. 5 (May): 47-72.
- Miles, T.I. and G.M. Lane, Hierarchical Modelling of Banks using ISI and SIMAN, in *Proceedings of ESM '91, the 1991 European Simulation Multiconference*, SCS Europe, Ghent, Belgium, June 1991.
- Murgiano, C., A Tutorial on WITNESS, in *Proceedings of 1990 Winter Simulation Conference*, eds. O. Balci, R.P. Sadowski and R.E. Nance, New Orleans, LA, SCS, San Diego, CA, pp177-179, Dec. 9-12, 1990.
- Nolan, P.J., Fegan, J.M. and Lane, G.M., An AI interface for the Engineering use of Simulation Software, in *Proceedings of the Fifth International Conference of the Irish Manufacturing Committee*, Belfast, Ireland, Sept. 1988.
- Norman, V.B., AutoMod II, in *Proceedings of 1990 Winter Simulation Conference*, eds. O. Balci, R.P. Sadowski and R.E. Nance, New Orleans, LA, SCS, San Diego, CA, pp94-98, Dec. 9-12, 1990.
- Pegden, C.D., R.E. Shannon, and R.P. Sadowski, *Introduction to Simulation using SIMAN*, McGraw-Hill Inc., Hightstown, NJ, 1990.
- Pritsker, A.A.B., *Introduction to Simulation and SLAM-II*, 3rd Edition, Halsted Press, John Wiley and Sons, NY, 1986.
- Schriber, T.J., Perspectives on Simulation using GPSS, in *Proceedings of 1990 Winter Simulation Conference*, eds. O. Balci, R.P. Sadowski and R.E. Nance, New Orleans, LA, SCS, San Diego, CA, pp5-13, Dec. 9-12, 1990.
- Shannon, R.E., M. Mayer, and H.H. Adelsberger, Expert Systems and Simulation, *Simulation*, Vol. 44, No. 6, 1985.

## AUTHOR BIOGRAPHIES

**JOSEPH M. FEGAN** is employed by Extech Limited, Galway, Ireland, as Software Manager, where he is involved in the development of interactive graphical tools for discrete event simulation. His current research interests include the use of AI code generation techniques in discrete event simulation, Object Oriented Programming, AI based simulation languages and hierarchical graphical simulation models.

**GEARÓID M. LANE** is Development Manager with Extech Limited, Galway, Ireland, where he works on the development of interactive graphics software for discrete event simulation. His current research interests include the application of AI knowledge techniques to discrete event simulation, the use of modified Petri nets in modelling of manufacturing systems, Object Oriented Programming and interactive computer graphics in simulation.

**PAUL J. NOLAN** is currently a Senior Lecturer in Mechanical Engineering at University College Galway, Ireland, where his research interests are in AI based simulation of discrete event and continuous time systems.